2° 动态规划

2023012173 赵奎睿

第一章

一、名词解释

1、操作系统：是计算机硬件上的第一层软件，用于管理硬件设备，提高利用率和吞吐量，便于用户使用。

4、实时操作系统：一种能在严格时间约束下完成特定任务的操作系统。

5、互斥共享：指多个进程访问共享资源时，必须以互斥的方式访问，以避免资源冲突。

三、填空

1、硬件系统、软件系统。

3、硬件资源，软件资源。

5、并发性，共享性

第二章

一、名词解释

3、临界区：指进程中访问共享资源的代码段，在*其之内同一时间只有一种个进程执行。

4、进程同步：指通过某种机制协调多个进程的执行顺序，以确保其不会发生冲突。

(1)基本要素：
1)最优子结构

三、填空

1、进程特征包括其结构特征，动态性，并发性、独立性和异步性。

5、比较进程和程序时，进程是动态概念，程序是静态概念。

6、由共享公有资源而造成的对并发~~程序~~进程执行速度的制约称为间接相互制约。

8、进程通信包括~~直接通信，间接通信和基类~~共享存储器系统，消息传递系统和管道通信系统

9、线程的基本状态包括就绪，执行和阻塞状态。

五、问答

2、 S1 → S2, S3
S2 → S4, S5
S3 → S6
S4, S5, S6 → S7

```
semaphore S1=1, S2=0, S3=0, S4=0, S5=0, S6=0, S7=0;
void P1(){
  wait(S1);
  执行S1
  signal(S2);
  signal(S3);
}
void P2(){
  wait(S2);
  执行S2
  signal(S4);
  signal(S5);
}
void P3(){
  wait(S3);
  执行S3
  signal(S6);
}
void P4(){
  wait(S4);
  signal(S7);
}
void P5(){
  wait(S5);
  执行S5
  signal(S7);
}
void P6(){
  wait(S6);
  执行S6
  signal(S7);
}
void P7(){
  signal(S7);
}
```

(1) 基本要素:

1) 最优子结构

2) 重叠子问题

mutex = 1, apple = 0; orange = 0, empty = 5;

```
void father(){
 while(true){
  String fruit = produce_fruit();
  wait(empty);
  wait(mutex);
  if(fruit == "apple"){
   apple++;}
  else {
   orange++;
  }
  signal(mutex);
 }
}
```

```
void son(){
  while(true){
   wait(orange);
   wait(mutex);
    吃;
   signal(mutex);
   signal(empty);
  }
}
```

```
void daughter(){
  while(true){
   wait(apple);
   wait(mutex);
   吃
   signal(mutex);
   signal(empty);
  }
}
```

4. semaphore mutex1 = 1, mutex2 = 1, empty1 = 1, empty2 = 1, full1 = 0, full2 = 0;

```
void PA(){
 while(true){
   wait(mutex1);
   wait(empty1);
   写;
   signal(mutex1);
   signal(full1);
 }
}
```

```
void PB(){
  while(true){
   wait(full1)
   wait(mutex1)
    读;
   signal(mutex1);
   wait(empty2);
   wait(mutex2);
   复制;
   signal(mutex2);
   signal(full2);}}
```

```
void PC(){
  while(true){
   wait(full2);
   wait(mutex2);
   打印;
   signal(mutex2);
   signal(empty2);
  }
}
```

signal(empty1)

5. semaphore mutex1=1, customers=0, barber=0;
chairs=N;, mutex2=1;

```
void customer(){
  while(true){
  wait(chairs);
  wait(mutex1);
  customers++;
  ~~wait(~~
  signal(mutex1);
  signal(barber);
  ~~wait(mutex2);~~ wait(customers);
                    wait(mutex2);
  理发;
  付款;
  signal(mutex2);
  signal(chairs);
  }
}
```

```
void barber(){
  while(true){
  wait(customers);
  wait(mutex2);
  customers--;
  signal(mutex2);
  signal(barber);
  }
}
```

第三章

一、名词解释

2. 处理机调度：指操作系统根据一定策略将CPU分配给进程，以实现进程的并发运行。

3. 周转时间：指作业行从提交到完成的总时间。

4. 死锁：因多个进程对资源的竞争而导致无法继续执行的状态。

三、填空

2. 作业在其生存时间内会经历提交，后备，执行和

6. 死锁的处理方法包括预防，避免，检测和

五、问答

1. (1) FCFS

顺序：ABCDE　平均周转 8.2　平均带权周转 2.567

(2) SJF

顺序：ABECD　平均周转 7.6　平均带权周转 1.84

(3) HRRN

顺序：ABCED　平均周转 8　平均带权 2.14

(4) 时间片轮转

顺序 AECBD　平均周转 10　平均带权 2.71

2. (1) A = 20 - 0 - 10 = 10ms

B = 50 - 0 = 10 = 40ms

C = 50 - 0 - 15 = 35ms

先执行A

(2) B = 50 - 10 - 10 = 30ms

C = 50 - 10 - 15 = 25ms

再执行C

(3) 最后执行B

3. (1) request 矩阵：

Allocation

0	0	3	2
1	0	0	0
1	3	5	4
0	3	3	2
0	0	1	4

安全序列 $\{P_0, P_3, P_4, P_1, P_2\}$ 安全

(2) ①request ≤ need

② request ≤ available

③ 分配后有无安全序列，不安全

④ 停止分配，收回资源

(3) 不会，当有其它 request 提出且大于 available 时才发生